Single-Filter Finite Fault Detection and Exclusion Methodology for Real-Time Validation of Plug-and-Play Sensors

JUAN JURADO
U.S. Air Force Test Pilot School, Edwards AFB, Edwards, CA, USA

JOHN RAQUET
IS4S-Dayton, Integrated Solutions for Systems, Inc. Beavercreek, OH, USA

CHRISTINE M. SCHUBERT KABBAN
Air Force Institute of Technology, Wright-Patterson AFB, Dayton, OH, USA

All-source navigation has become increasingly relevant over the past decade with the development of viable alternative sensor technologies. However, as the number and type of sensors informing a system increases, so does the probability of corrupting the system with sensor modeling errors, signal interference, and undetected faults. Though the latter of these has been extensively researched, the majority of existing approaches have designed algorithms centered around the assumption of simultaneously redundant, synchronous sensors with well-understood measurement models, none of which are guaranteed for all-source systems. As part of an overall all-source assured or resilient navigation objective, this research contributes a key component—validation of sensors which have questionable sensor models, in a fault-agnostic and sensor-agnostic manner, and without compromising the ongoing navigation solution in the process. The proposed algorithm combines a residual-based test statistic with the partial update formulation of the Kalman–Schmidt filter to provide a reliable method for sensor model validation that protects the integrity of the navigation solution during the validation process, all using only a single existing filter. The performance of the proposed method is validated against traditional fault detection and exclusion methods (such as normalized solution separation and conventional residual sequence monitoring) using Monte–Carlo simulations in a 2D non-Global Positioning System navigation problem with a plug-and-play position sensor.

Manuscript received June 27, 2019; revised October 23, 2019 and March 26, 2020; released for publication July 5, 2020. Date of publication July 21, 2020; date of current version February 9, 2021.

DOI. No. 10.1109/TAES.2020.3010394

Refereeing of this contribution was handled by Z. Davis.

This work was supported by U.S. Air Force.

Authors' addresses: Juan Jurado is with the U.S. Air Force Test Pilot School, Edwards AFB, Edwards, CA 93524 USA, E-mail: (jdjurado@gmail.com); John Raquet is with the IS4S-Dayton, Integrated Solutions for Systems, Inc., Beavercreek, OH 45324 USA, E-mail: (john.raquet@is4s.com); Christine M. Schubert Kabban is with the Department of Mathematics and Statistics, Air Force Institute of Technology, Wright-Patterson AFB, Dayton, OH 45433 USA, E-mail: (christine.schubert@afit.edu). *(Corresponding author: Juan Jurado.)*

I. INTRODUCTION

All-source navigation and Assured Position Navigation and Timing (APNT) have become increasingly relevant over the past two decades, as the research community continues to mature sensor technologies (e.g., vision [1], radio [2], magnetic [3], etc.) and integrate them into navigation systems [4]. However, each additional sensor allowed into a navigation system creates an opportunity for corrupting the navigation solution with errors in sensor modeling, unexpected signal interference, or undetected sensor faults. Though the latter of these challenge areas has been extensively researched [5]–[20], the primary objective has traditionally been to provide navigation solution integrity (via fault detection and exclusion) in an ongoing multisensor navigation process, with the assumption that each sensor in the system is equally likely to experience a fault, and that sensors are properly modeled at the start of the navigation process. Additionally, the research in this challenge area has focused almost exclusively on simultaneously redundant and synchronous multisensor systems such as the satellites in the Global Positioning System (GPS) constellation.

Our overall research motivation was to address the APNT challenge for all-source navigation by creating a general, fault-agnostic resilient navigation framework, which is described in [21]. The Autonomous and Resilient Management of All-source Sensors (ARMAS) framework provides APNT through the online or real-time detection, exclusion, and self-correction (i.e., autotuning) of sensor models that do not match observed measurements, where a biased sensor is simply one possible model mismatch. In support of this overall objective, two specific all-source research areas were investigated for the aforementioned framework: (1) the ability to monitor online/trusted sensors (i.e., sensors currently informing the navigation solution) for fault detection and exclusion, and integrity computations, and (2) the ability to initialize offline plug-and-play sensors by validating their stated measurement models while protecting the navigation solution. Of these two processes, the former is partly addressed in the previously mentioned research for systems like GPS, and in [22], for all-source, non-synchronous sensors, and sensor model mismatches beyond biases. The latter, which we are referring to as the sensor validation problem, is virtually unaddressed (at the time of writing) in current research and therefore constitutes the focus of this article.

II. BACKGROUND

In the context of our research, sensor validation refers to the process of initializing an offline plug-and-play sensor

into an ongoing navigation system that is already being informed by a set of (previously initialized) online and trusted sensors, and determining whether or not its measurements are statistically adhering to their stated measurement model. The set of online sensors are presumably being monitored for fault detection and exclusion using one of the many integrity monitoring methods previously discussed, such as [11] and [22]. In this sense, the offline sensor is initialized separately due to the presumption of possible errors in sensor modeling or the presence of sensor faults. This concept also encompasses the possibility of reinitializing a sensor that was previously found "faulty" and placed offline by a multisensor integrity-monitoring process (e.g., due to temporary interference or model changes due to environmental variables). Therefore, the challenge in sensor validation is not focused around continuous integrity monitoring for the duration of the navigation sequence, but rather during a fixed "initialization" period, after which the sensor is either determined valid and placed into the integrity-monitoring pool with all other online sensors, or deemed invalid and either placed back into offline status or placed into sensor model remedial measures as described in [21]. The primary challenge in validating an offline sensor is ensuring its (presumed) faulty measurements do not corrupt the ongoing navigation solution, while simultaneously ensuring the chosen statistical test is capable of detecting nuanced differences between the observed measurements and their stated measurement model [23].

The online sensor validation process can then be posed as a finite-time Fault Detection and Exclusion (FDE) task with a focus on a single new (untrusted) sensor whose measurement model includes sensor-unique states that are not currently estimated by the filter or shared by other sensors in the system. The end-goal of this validation is to autonomously make a trusted/not-trusted decision using a finite validation period. To accomplish this, one could employ existing multiple-filter solution separation-based [11] or residual-based [22] FDE algorithms by simply adding the plug-and-play sensor into the pool of trusted online sensors, and allowing the FDE algorithm to determine if the sensor is faulty. The main drawback from using these approaches is that they require increasing the number of subfilters needed to accommodate the additional sensor, and since the trustworthiness of the sensor model is already in question, these approaches could allow for degradation of the navigation solution between the time the sensor is "plugged in" and the time a fault is detected. Additionally, these approaches require the use of subfilters, which are computationally expensive. Focusing on a single-filter system, one could employ conventional residual sequence-monitoring methods [17], [24] using a finite sequence length for fault detection, but without fault exclusion. In [19], a single-filter residual sequence FDE method is proposed without the use of subfilters. However, the method's fault exclusion depends on the detector's ability to sense the fault prior to reconstructing the fault-excluded solution using solution separation statistics. Additionally, the usefulness of this type of method on nonredundant multi-sensor systems with multidimensional measurement vectors (i.e., all-source systems) is not readily apparent.

Given none of the existing methods meet our desired functionality, our motivation is then to develop a single-filter methodology for finite-length FDE that is readily compatible with heterogeneous all-source plug-and-play sensors. As such, the method developed in this article, referred to henceforth as Real-time Validation for Plug-and-play Sensors (RVPS), provides a significant contribution to the state-of-the-art in that it formally introduces and defines the concept of plug-and-play sensor validation, and provides one generalized method for plug-and-play sensor model validation that meets the aforementioned requirements.

The remainder of this article is divided into three additional sections. Section III develops the necessary filtering notation, describes the concept of the Kalman–Schmidt "partial update" as a key enabler for the proposed method, and derives the residual-based sum of Chi-square test statistic used for validation. Section IV compares fault detection performance across a variety of multifilter and single-filter FDE approaches using Monte–Carlo simulations on a 2D all-source plug-and-play navigation problem. Finally, Section V summarizes the research contributions, provides ideas for future work and concludes the article.

III. METHODOLOGY

Adapting the Kalman filter [25] notation from [24], [26], consider an ongoing (possibly) nonlinear dynamic system of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \mathbf{G}(t)\mathbf{w}(t) \tag{1}$$

where $\mathbf{x}$ is the $N \times 1$ navigation state vector containing the system states, $\mathbf{u}$ is the control input vector, $\mathbf{G}$ is an $N \times W$ linear operator, and $\mathbf{w}$ is a $W \times 1$ white Gaussian noise process with a $W \times W$ continuous process noise strength matrix $\mathbf{Q}$. At time $t = t_k$, the state estimate vector and corresponding state estimation error covariance matrix are given by $\hat{\mathbf{x}}(t_k)$ and $\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_k)$, respectively, and produced by the system's (potentially only) "main filter," which is informed by a set of online (trusted) sensors that are presumably being monitored for fault detection and exclusion. Next, starting at a given initialization time, we wish to begin validating an offline and untrusted plug-and-play sensor that provides (possibly) multidimensional $Z \times 1$ measurements of the form

$$\mathbf{z}(t_k) = \mathbf{h}\left[\mathbf{x}(t_k), \boldsymbol{\epsilon}(t_k), \mathbf{u}(t_k), t_k\right] + \mathbf{v}_k \tag{2}$$

where $\mathbf{h}$ is a (possibly) nonlinear measurement function, $\boldsymbol{\epsilon}$ is a $U \times 1$ vector of sensor-unique states needed for measurement processing, not currently estimated by the filter, and observable only by the sensor in question, and $\mathbf{v}_k$ is a $Z \times 1$ discrete white Gaussian noise process with covariance matrix $\mathbf{R}(t_k)$. In order to initialize the sensor and process measurements, the sensor-unique states, $\boldsymbol{\epsilon}$, must be augmented into $\mathbf{x}$ using their initial estimate $\hat{\boldsymbol{\epsilon}}(t_i^-)$ and corresponding initial estimation error covariance matrix, $\mathbf{P}_{\hat{\boldsymbol{\epsilon}}\hat{\boldsymbol{\epsilon}}}(t_i^-)$, where t_i is the initialization time. After augmenting the state-space, we wish to detect sensor faults during the finite

validation period while ensuring the newly initialized sensor does not corrupt the ongoing navigation solution. Focusing on a single-filter system excludes the use of subfilters, which prevents the use of a solution separation detector as well as the associated fault-exclusion capability. This leaves one option, which is to perform residual sequence monitoring without applying the measurement updates. Though this so-called no-update residual sequence monitor prevents faulty measurements from corrupting the navigation solution, it creates a new problem of correlation among the samples in the residual sequence, which invalidates the assumption that the sum of residual norms follows a central Chi-square distribution [6], [24], [27]. This problem is exacerbated when the sensor-unique states, $\boldsymbol{\epsilon}$, have long correlation time constants, with the worst case being a constant bias (i.e., infinite time constant).

We solve the correlation problem using two modifications to the no-update residual sequence detector. First, we employ the partial update [28] formulation of the Kalman–Schmidt filter [29] in order to eliminate the correlation in the residual sequence due to $\boldsymbol{\epsilon}$. The Kalman–Schmidt filter formulation enables us to designate a subset of the state-space variables as "consider" states whose statistical distribution is considered during the measurement update, but whose distribution (e.g., the state estimate and error covariance matrix) is unaffected by the measurement update. Though traditionally the states designated as "consider" states have been primarily unobservable biases, in this application, we designate the "core" navigation solution states (e.g., position, velocity, acceleration, etc.) as "consider" in order to guarantee a fault-free single-filter solution even while estimating sensor-unique states, $\boldsymbol{\epsilon}$. Allowing the filter to estimate $\boldsymbol{\epsilon}$ via the partial update eliminates the correlation among residuals caused by the unobservable states $\boldsymbol{\epsilon}$. However, the correlation among residuals caused by not incorporating Sensor B measurements into the core navigation states, $\mathbf{x}$, still remains. We eliminate this remaining correlation by optimizing the residual sequence sample spacing within the finite validation period using existing filter covariance kernel analysis methods [30], [31] as summarized in a later section.

A. Executing the Partial Update

We begin the validation process at $t = t_i$ by augmenting the existing state-space with the initial estimate and initial error covariance matrix of the sensor-unique states, $\boldsymbol{\epsilon}$, using

$$\hat{\mathbf{y}}(t_i^-) = \left[\hat{\mathbf{x}}(t_i^-)\ \hat{\boldsymbol{\epsilon}}(t_i^-)\right]^{\mathrm{T}} \tag{3}$$

$$\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_i^-) = \begin{bmatrix} \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_i^-) & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\hat{\boldsymbol{\epsilon}}\hat{\boldsymbol{\epsilon}}}(t_i^-) \end{bmatrix} \tag{4}$$

where $\hat{\mathbf{y}}$ is the $(N+U)\times 1$ augmented state estimate, $\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}$ is the corresponding $(N+U)\times(N+U)$ augmented state estimation error covariance matrix, and keeping in mind the state dynamics associated with the propagation of $\boldsymbol{\epsilon}$ must also be augmented into $\mathbf{G}$, $\mathbf{w}$, and $\mathbf{Q}$. Next, we define the partial update vector, $\boldsymbol{\beta}$, and its corresponding partial update matrix, $\mathbf{B}$, using

$$\underset{(N+U)\times 1}{\boldsymbol{\beta}} = \begin{bmatrix} \underset{1\times N}{\mathbf{0}} & \underset{1\times U}{\mathbf{1}} \end{bmatrix}^{\mathrm{T}} \tag{5}$$

$$\boldsymbol{\gamma} = \mathbf{1} - \boldsymbol{\beta} \tag{6}$$

$$\boldsymbol{\Gamma} = \boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathrm{T}} \tag{7}$$

$$\mathbf{B} = \mathbf{1} - \boldsymbol{\Gamma} \tag{8}$$

where the entries in $\boldsymbol{\beta}$ corresponding to $\mathbf{x}$ are set to zero, and the entries corresponding to $\boldsymbol{\epsilon}$ are set to one. When the first sensor measurement becomes available at time $t = t_k \geq t_i$, we first compute a standard (i.e., full) measurement update to obtain the postupdate augmented state estimate, $\hat{\mathbf{y}}(t_k^+)$, and its corresponding error covariance matrix, $\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_k^+)$ using

$$\mathbf{y}(t_k^+) = \mathbf{y}(t_k^-) + \mathbf{K}(t_k)\mathbf{r}(t_k) \tag{9}$$

$$\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_k^+) = [\mathbf{I} - \mathbf{K}(t_k)\mathbf{H}(t_k^-)]\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_k^-) \tag{10}$$

where $\mathbf{H}(t_k^-)$ represents the Jacobian of $\mathbf{h}$ about the point $\hat{\mathbf{y}}(t_k^-)$, and the Kalman gain, $\mathbf{K}(t_k)$, and the preupdate residual, $\mathbf{r}(t_k)$, are given by

$$\mathbf{K}(t_k) = \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k^-)\mathbf{P}_{\mathbf{rr}}(t_k)^{-1} \tag{11}$$

$$\mathbf{r}(t_k) = \mathbf{z}(t_k) - \hat{\mathbf{z}}(t_k^-) \tag{12}$$

and the the estimated measurement, $\hat{\mathbf{z}}(t_k^-)$, its covariance matrix, $\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k^-)$, and the preupdate residual covariance matrix, $\mathbf{P}_{\mathbf{rr}}(t_k)$, are given by

$$\hat{\mathbf{z}}(t_k^-) = \mathbf{h}\left[\hat{\mathbf{y}}(t_k^-), \mathbf{u}(t_k), t_k\right] \tag{13}$$

$$\mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k^-) = \mathbf{H}(t_k^-)\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_k^-)\mathbf{H}^{\mathrm{T}}(t_k^-) \tag{14}$$

$$\mathbf{P}_{\mathbf{rr}}(t_k) = \mathbf{R}(t_k) + \mathbf{P}_{\hat{\mathbf{z}}\hat{\mathbf{z}}}(t_k^-). \tag{15}$$

Here it is important to note the measurement update equations described above are based on the Extended Kalman Filter (EKF) algorithm. For more information in computing the same in an Unscented Kalman Filter (UKF), the reader is referred to [32] and [33]. Next, as shown in [28], the partial update is computed by modifying the updated state estimate vector from (9) and its corresponding error covariance matrix from (10), using (5)–(8), and

$$\hat{\mathbf{y}}(t_i^{++}) = \boldsymbol{\beta} \odot \hat{\mathbf{y}}(t_i^+) + \boldsymbol{\gamma} \odot \hat{\mathbf{y}}(t_i^-) \tag{16}$$

$$\mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_i^{++}) = \mathbf{B} \odot \mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_i^+) + \boldsymbol{\Gamma} \odot \mathbf{P}_{\hat{\mathbf{y}}\hat{\mathbf{y}}}(t_i^-) \tag{17}$$

where $\odot$ is the Hadamard or point-wise product. The partial update process described above can be then repeated at every epoch during the user-defined finite validation period.

B. Optimizing the Residual Sequence Spacing

As previously discussed, the partial update formulation eliminates part of the residual correlation found in the no-update residual sequence detector, which protects the navigation solution during the validation process. The remaining residual correlation is caused by not incorporating Sensor B measurements into the core navigation states, $\mathbf{x}$. This correlation can be eliminated or minimized through the analysis of the filter's covariance kernel, which can be done online prior to initializing the plug-and-play sensor, or even offline for linear systems with constant $\mathbf{H}$ and $\mathbf{R}$

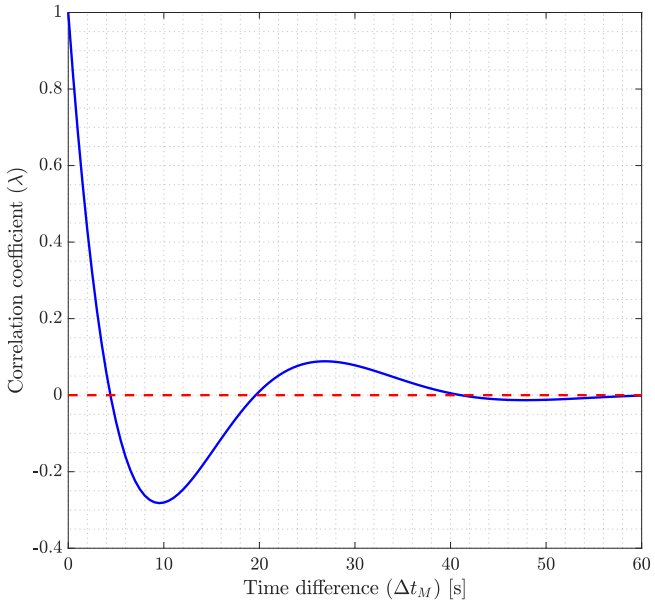

Fig. 1. Example filter covariance kernel analysis.

terms. This process is described in detail in [30], [31], and briefly summarized here for completeness.

Under conventional (full update) circumstances, the residual sequence from (12) has been proven to be white [26]. As shown in [26], this is due to the independence between the current residual, $\mathbf{r}(t_k)$, and all previous measurements up to $\mathbf{z}(t_{k-1})$. However, in a partial update sequence, the innovations from the previous measurements have only been incorporated into a subset of the state estimate, $\hat{\mathbf{y}}(t_k^-)$, namely the sensor-unique states, $\hat{\boldsymbol{\epsilon}}(t_k^-)$. As such, the residual sequence now follows the correlation of the state estimation error sequence, $\mathbf{e}(t_k^+)$, which is given by

$$\mathbf{e}(t_k^+) = \mathbf{y}(t_k) - \hat{\mathbf{y}}(t_k^+) \tag{18}$$

as projected onto the untrusted sensor's measurement space. In other words, the time correlation of the partial update residual sequence is driven by the time correlation of the filter's state estimation error sequence projected onto measurement space. To characterize this correlation, we begin by defining the covariance kernel, $\mathbf{C}$, for a given propagate–update epoch using the existing (trusted) sensor models $\mathbf{H}_T$ and Kalman gains, $\mathbf{K}_T$, in

$$\mathbf{C}(t_{k+1}) = [\mathbf{I} - \mathbf{K}_T(t_k)\mathbf{H}_T(t_k)]\,\Phi(t_k)\mathbf{C}(t_k) \tag{19}$$

where $\mathbf{I}$ is an appropriately sized identity matrix, and Φ is the state transition matrix given the filter's propagation model. Starting with $\mathbf{C}(t_0) = \mathbf{I}$, we can recursively use (19) to compute the dominant correlation coefficient, $\lambda(t_k)$, as a function of $\Delta t_M = t_k - t_0$ by projecting $\mathbf{C}(t_k)$ into the untrusted sensor's measurement space using the untrusted measurement model $\mathbf{H}_U$ in

$$\mathbf{D}(t_k) = \mathbf{H}_U(t_k)\mathbf{C}(t_k)\mathbf{H}_U(t_k)^{\mathrm{T}} \tag{20}$$

then setting $\lambda(t_k)$ equal to the largest (absolute) eigenvalue of $\mathbf{D}(t_k)$. Fig. 1 illustrates a sample history of $\lambda(t_k)$ as a function of Δt_M across a 60 s finite sequence length. Given a sequence length corresponding to the user-defined validation period, one can use any optimization routine to maximize the number of samples in the residual sequence while minimizing their intercorrelation using the computed kernel in Fig. 1 as a cost function.

C. Computing the Residual-Based Test Statistic

Having defined the partial update and residual spacing mechanisms, we now focus on defining the residual-based test statistic. Since our overall research objective was to limit the assumptions on the type of fault (i.e., the fault could be a bias, an incorrectly stated noise covariance matrix, or incorrect calibration of measurement function parameters), we did not model two competing distributions as would be needed to employ a Likelihood Ratio Test (LRT) [34]. Instead, we focused on the distribution resulting from summing the squared Mahalanobis distance [27] across a sequence of preupdate residuals, and selecting a threshold based on a desired probability of false alarm, P_f.

Given a Z-dimensional Gaussian distribution with mean $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}$, the squared Mahalanobis distance, d^2, between an observation, $\mathbf{g}$, and the centroid of the distribution is then given by

$$d^2 = (\mathbf{g} - \boldsymbol{\mu})^{\mathrm{T}}\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{g} - \boldsymbol{\mu}). \tag{21}$$

Additionally, d^2 is known [27], [35] to follow a central Chi-square distribution with Z degrees of freedom. Moreover, the sum of M *independent* d^2 distances is also known to follow a central Chi-square distribution with $M \times Z$ degrees of freedom. As previously discussed, Kalman filter preupdate residuals (i.e., innovations) form a white sequence [11], [26] as long as the measurement updates are incorporated into the navigation solution. In the case of the partial update, the time-spacing of residual samples, Δt_M, must be optimized to minimize the correlation among the residuals. The effects of applying a partial update on residual correlation are illustrated and discussed in Section IV. Assuming the residual sequence forms a zero-mean white sequence, we can let $\mathbf{g} = \mathbf{r}(t_k)$ from (12), $\boldsymbol{\Sigma} = \mathbf{P}_{\mathbf{rr}}(t_k)$ from (15), and $\boldsymbol{\mu} = \mathbf{0}$. Subsequently, we can compute the fault detection test statistic, χ^*, using

$$\chi^* = \sum_{s=k}^{k+M-1} \mathbf{r}^{\mathrm{T}}(s\Delta t_M)\,[\mathbf{P}_{rr}(s\Delta t_M)]^{-1}\,\mathbf{r}(s\Delta t_M) \tag{22}$$

where Δt_M is the optimized residual sample spacing. A fault is then declared if

$$\chi^* > \chi^2(1-\alpha, M \times Z) \tag{23}$$

where α is the desired P_f.

IV. SIMULATION RESULTS

The proposed method was evaluated via a series of Monte–Carlo simulations using four vehicles informed by an online trusted 2D position sensor (Sensor A) and an offline untrusted 2D position sensor (Sensor B). In the first vehicle (Aircraft 1), RVPS was used to initialize and validate Sensor B using the proposed method (i.e., using

a properly spaced partial update sequence detector). In the second vehicle (Aircraft 2), Sensor B was validated via a multifilter horizontal position solution separation detector as implemented in [11]. In the third vehicle (Aircraft 3), the residual-based test statistic developed in this article was used, however with no updates, meaning Sensor B measurements were not allowed to affect any states in the filter solution. Conversely, in the fourth vehicle (Aircraft 4), a conventional residual sequence monitor was used, meaning the measurements from Sensor B were allowed to update all the states in the filter solution during the collection of the residual test statistic. For all simulations, the true system dynamics were driven by a 2D kinematic model given by

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}_p(t) \\ \dot{\mathbf{x}}_v(t) \\ \dot{\mathbf{x}}_a(t) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_v(t) \\ \mathbf{x}_a(t) \\ -\frac{1}{\tau_a}\mathbf{x}_a(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{w}(t) \end{bmatrix} \tag{24}$$

where $\mathbf{x}_p$ is the vehicle's 2D position in [m], $\mathbf{x}_v$ is the 2D velocity in [m/s], $\mathbf{x}_a$ is the 2D acceleration in m/s^2 and propagated by a First Order Gauss-Markov (FOGM) process with time constant $\tau_a = 10$ s, and variance $\sigma_a = 0.01^2$ m^2/s^4, making $\mathbf{w}(t)$ a 2D white Gaussian noise process with $E[\mathbf{w}(t)\mathbf{w}(t+\tau)^{\mathrm{T}}] = \mathbf{Q}\delta(\tau)$ and

$$\mathbf{Q} = (1.5 \times 10^{-3})^2 \underset{2\times 2}{\mathbf{I}}\ \mathrm{m}^2/\mathrm{s}^2. \tag{25}$$

Sensor A measurements were modeled using

$$\mathbf{z}^{[A]}(t_k) = \mathbf{x}_p(t_k) + \mathbf{v}_k^{[A]} \tag{26}$$

$$E\left[\mathbf{v}_k^{[A]}\mathbf{v}_k^{[A]^{\mathrm{T}}}\right] = \mathbf{R}^{[A]}(t_k) = \begin{bmatrix} 100^2 & 0 \\ 0 & 100^2 \end{bmatrix} \mathrm{m}^2 \tag{27}$$

and its simulated measurements were drawn from the modeled distribution. Meanwhile, Sensor B measurements were modeled as

$$\mathbf{z}^{[B]}(t_k) = \mathbf{x}_p(t_k) + \begin{bmatrix} b \\ b \end{bmatrix} + \mathbf{v}_k^{[B]} \tag{28}$$

$$E\left[\mathbf{v}_k^{[B]}\mathbf{v}_k^{[B]^{\mathrm{T}}}\right] = \mathbf{R}^{[B]}(t_k) = \begin{bmatrix} 1^2 & 0 \\ 0 & 1^2 \end{bmatrix} \mathrm{m}^2 \tag{29}$$

where b is a constant (for each trial) but unknown turn-on bias affecting both the x- and y-dimensions equally. The simulations consisted of 10 000 trials where the measurements from Sensor B were drawn from the modeled distribution, and an additional 10 000 trials where the measurements were corrupted with an unmodeled x-position bias of 10 m. For each trial, the initial state estimation error covariance matrix was set to

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_0) = \mathrm{diag}\left(\begin{bmatrix} 1 & 1 & 1 & 1 & 0.01^2 & 0.01^2 \end{bmatrix}\right) \tag{30}$$

while the initial state estimate, $\hat{\mathbf{x}}(t_0)$, was set equal to the true initial state, which was drawn from a $\mathcal{N}(\mathbf{0}, \mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}(t_0))$ distribution. Each trial was propagated using $\Delta t_k = 0.5$ s, starting at $t_k = 0$ s with an offline sensor initialization at $t_i = 60$ s. The sensor validation period was set to 60 s, and all trials were terminated at $t_k = 150$ s. At t_i, the constant Sensor B turn on bias, b, was drawn from a $\mathcal{N}(0, 100^2\mathrm{m}^2)$ distribution, its initial estimate was set to zero, and its initial estimation error covariance was set to 100^2 m^2.

For the vehicles using the partial or no-update detectors (Aircraft 1 and Aircraft 3), three different residual sequence spacings were used in order to demonstrate the effect of sample spacing on the resulting empirical χ^2 distribution. The first sequence spacing was defined by a single residual sample collection at $t = t_i$, making $M = 1$ in (22). Next, 19 was used to find an optimal residual spacing which maximized the number of samples while minimizing correlation. This process resulted in selecting $\Delta t_M = 20$ s, leading to $M = 3$ samples across the 60-s validation period. Finally, a third sequence spacing of $\Delta t_M = 10$ s (i.e., $M = 6$) was used to demonstrate the effects of suboptimal spacing on the resulting distribution. For the vehicle using horizontal position solution separation (Aircraft 2), the solution separation test was performed continuously (every update epoch) during the 60-s validation period. Finally, for the vehicle using conventional residual sequence monitoring (Aircraft 4), the residual sequence was spaced at the update rate $\Delta t = 0.5$ s, leading to $M = 120$ samples.

From a partial update perspective, we define the Sensor B turn-on bias, b, as the sensor-unique state, making $\boldsymbol{\epsilon} = b$. Using (5)–(8), at the time of Sensor B initialization, the 6×1 trusted navigation solution, $\mathbf{x}$, was augmented with the turn-on bias state, $\boldsymbol{\epsilon}$, to form the 7×1 augmented state vector, $\mathbf{y}$, resulting in

$$\boldsymbol{\beta} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

$$\boldsymbol{\gamma} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{32}$$

$$\underset{7\times 7}{\boldsymbol{\Gamma}} = \begin{bmatrix} \underset{6\times 6}{\mathbf{1}} & \vdots \\ \cdots & \mathbf{0} \end{bmatrix} \tag{33}$$

$$\underset{7\times 7}{\mathbf{B}} = \begin{bmatrix} \underset{6\times 6}{\mathbf{0}} & \vdots \\ \cdots & \mathbf{1} \end{bmatrix}. \tag{34}$$

Fig. 2 illustrates a sample trajectory from one of the Monte–Carlo trials with $M = 3$ for the partial and no-update detectors. As expected, the RVPS (Aircraft 1), solution separation (Aircraft 2), and no-update (Aircraft 3) detectors were all able to detect the 10-m bias via their respective test statistic. It is important to note, however, that in this particular instance, the solution separation detector did not immediately (i.e., the first sample after $t = t_i$) detect the fault, allowing the main filter solution in Aircraft 2 to drift with biased measurements until the fault was detected and excluded (by swapping the main filter solution with the subfilter excluding Sensor B). This behavior is not observed in Aircraft 1 and Aircraft 3 since both the proposed RVPS (partial update) detector and the no-update detector prevent the sensor being validated from affecting the core navigation solution during the validation process. Meanwhile, the conventional residual sequence detector (Aircraft 4) failed to detect the fault, allowing each subsequent biased Sensor B measurement to affect the navigation solution, thereby further inhibiting fault detection due to the absorption of the bias into the x-position state.

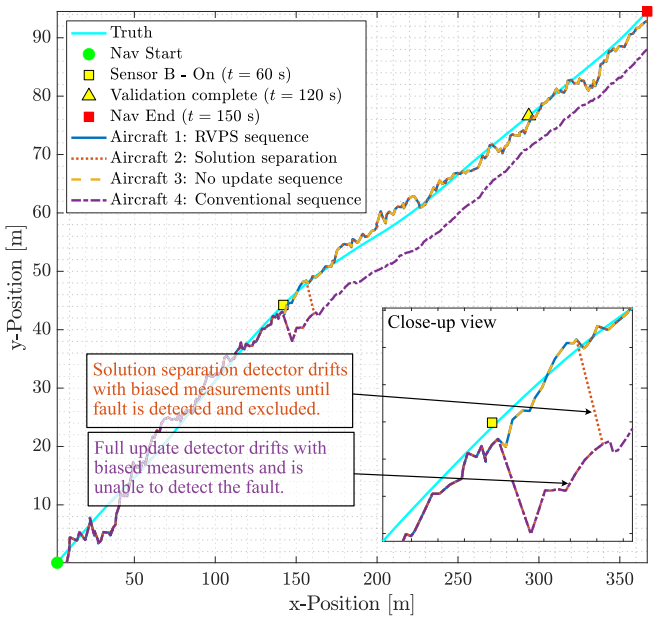

Fig. 2. Validation trajectory comparison, 10-m bias fault.

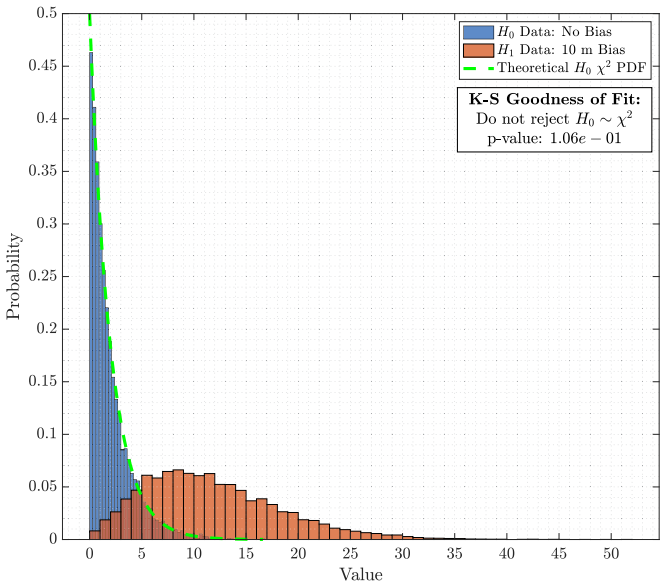

Fig. 4. Test statistic distributions: Solution separation.

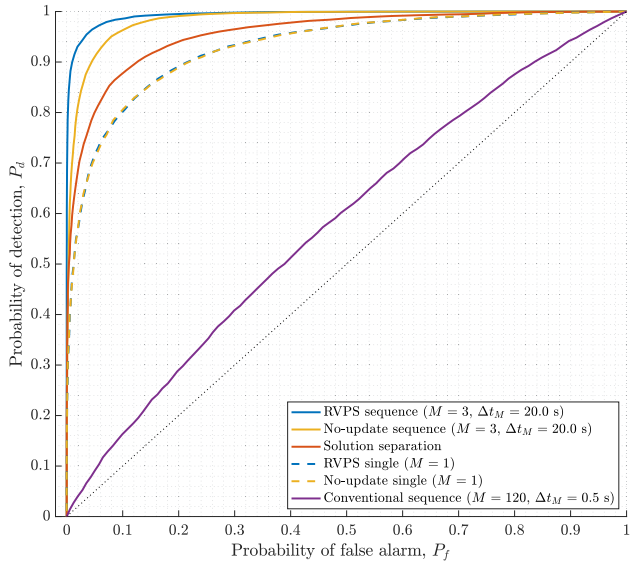

Fig. 3. Detection and false alarm comparison, 10-m bias fault.

Fig. 3 illustrates the probability of detection, P_d, versus the probability of false alarm, P_f, for a variety of detectors. As expected from the sample trajectory comparison, the conventional sequence detector performed poorly due to Sensor B's ability to affect all filter states, which allowed the filter solution to absorb the unmodeled x-position bias into the x-position state estimate. Next, both the RVPS detector and the no-update detector performed equally marginally when only one residual sample was used in the test statistic (i.e., $M = 1$). This is not surprising due to the relatively small size of the unmodeled bias, motivating the need for a residual sequence when using filtered residual-based fault detection. As expected, the three-sample ($M = 3$) no-update detector performed significantly better than its single-sample counterpart, roughly matching the performance of the horizontal position solution separation detector, which again highlights the advantage of using a residual sequence for detection. Finally, the three-sample ($M = 3$) RVPS detector performed significantly better than its single-sample counterpart, and outperformed all other detectors. Though the motivation for the proposed three-sample RVPS detector is clear based on detection performance, it is important to analyze the expected and actual test statistic distributions for each case in order to accurately predict P_f for the detector of choice.

Based on (22) and (23), the fault-free residual-based test statistics (i.e., RVPS, no-update, and conventional sequence) should follow a $\chi^2(M \times 2)$ distribution since Sensor B measurements were two-dimensional. Consequently, the expected fault-free distributions for our residual-based test statistics were $\chi^2(2)$, $\chi^2(6)$, and $\chi^2(12)$ for $M = 1$, $M = 3$, and $M = 6$, respectively. Additionally, given a horizontal (2D) position solution separation vector, the expected distribution of the solution separation test statistic was $\chi^2(2)$. Figs. 4–9 illustrate the actual and predicted fault-free test statistic distributions as well as the fault-present distributions for each of the detection schemes illustrated in Fig. 3. Each figure also displays a p-value for the Kolmogorov-Smirnov Test (KST) [36] where the goodness of fit between the empirical and theoretical H_0 distribution is tested. Note p-values lower than the significance level of the test (usually 0.05) lead to the rejection of the null hypothesis, or in this case, the conclusion that the empirical and theoretical distributions are not statistically equal.

As shown, the fault-free test statistic distributions of all except one detection method adhered to their theoretical values, allowing for accurate fault detection threshold selection based on a desired P_f. These results validate the accuracy of the simulations, and the whiteness of the residual sequences. The one exception was the no-update sequence ($M = 3$), which is illustrated in Fig. 9.

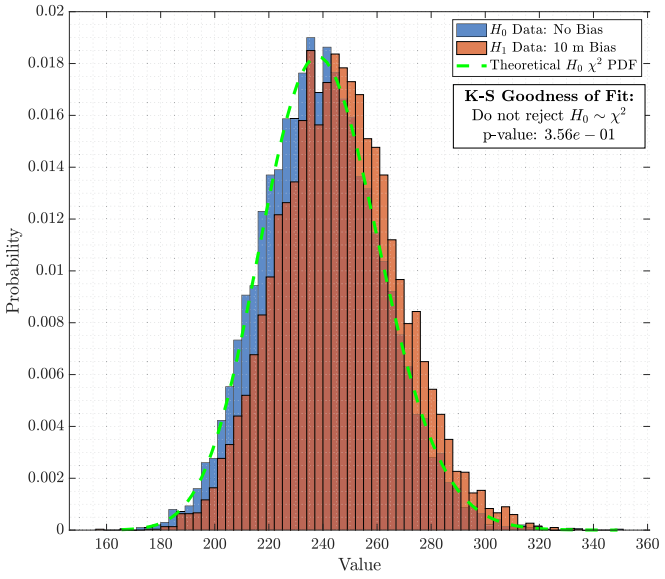

Fig. 5. Test statistic distributions: Conventional sequence ($M = 120$, $\Delta t_M = 0.5$ s).

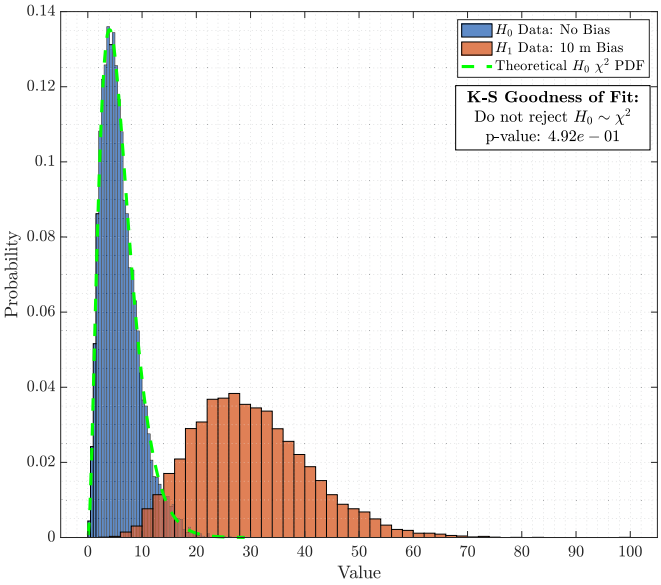

Fig. 7. Test statistic distributions: RVPS sequence ($M = 3$, $\Delta t_M = 20$ s).

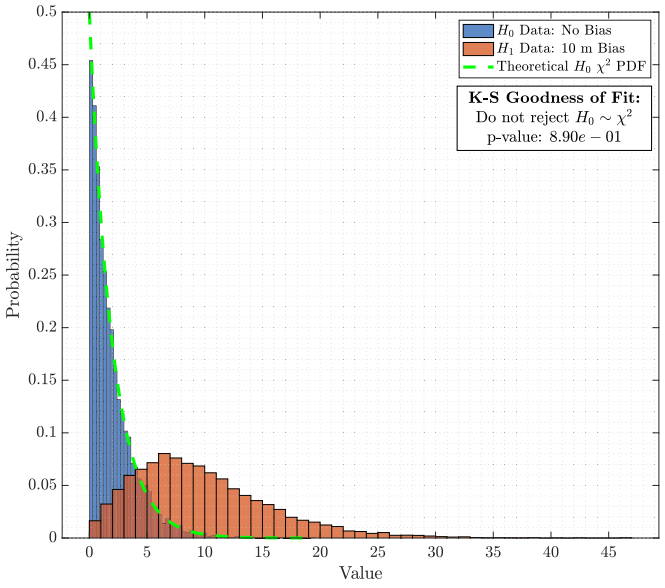

Fig. 6. Test statistic distributions: RVPS single ($M = 1$).

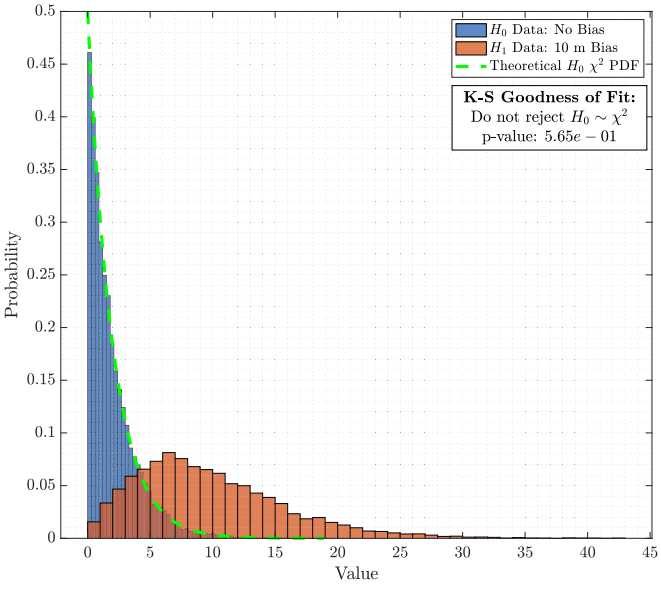

Fig. 8. Test statistic distributions: No-update single ($M = 1$).

Though this single-filter detector performed roughly equal to the multifilter solution separation method, its actual fault-free test statistic distribution was significantly different than expected, making threshold selection based on P_f unreliable. This apparent lack of goodness of fit is driven by residual correlation, even though the no-update sequence spacing was also derived using 19. This result supports the motivation for performing a partial update in the presence of sensor-unique states during validation. That is, since the no-update sequence detector did not estimate the sensor-unique state, b, which was modeled as a constant bias with 100^2 m^2 covariance, its residuals remained correlated even with the optimized 20-s sample spacing.

Finally, Table I summarizes the empirical P_f realized for a variety of detectors using a theoretical $P_f = 0.05$ threshold. Next, the empirical threshold needed to achieve the desired $P_f = 0.05$ is compared to the predicted threshold. Additionally, the previously mentioned KST p-value is listed for each detector type, with a low p-value indicating a lack of fit. The rows shaded in gray indicate detectors that exhibited a lack of fit as demonstrated by low p-values and significant differences between theoretical and actual P_f. As shown, the RVPS sequence detector with the optimal spacing ($M = 3$) exhibited desirable fit statistics, while increasing the number of samples up to $M = 6$ led to a lack of fit, due to increased residual sequence correlation. In contrast, and as previously discussed, the no-update sequence detector exhibited a lack of fit even with the optimized sequence spacing, highlighting the need to estimate sensor-unique states.

TABLE I
Goodness of Fit Summary for Detector H_0 Distributions

Detector type	Desired P_f	Actual P_f	Theoretical threshold	Empirical threshold	KS GOF p-value
Solution separation	0.05	0.0503	5.9915	5.9981	1.06×10^{-1}
Conventional sequence ($M = 120$)	0.05	0.0460	277.1376	276.4054	3.56×10^{-1}
RVPS single ($M = 1$)	0.05	0.0476	5.9915	5.9111	8.90×10^{-1}
RVPS sequence ($M = 3$)	0.05	0.0532	12.5916	12.7698	4.92×10^{-1}
RVPS sequence ($M = 6$)	0.05	0.0691	21.0261	22.5121	0.00
No-update single ($M = 1$)	0.05	0.0491	5.9915	5.9701	5.66×10^{-1}
No-update sequence ($M = 3$)	0.05	0.0958	12.5916	15.7629	0.00
No-update sequence ($M = 6$)	0.05	0.1365	21.0261	30.0677	0.00

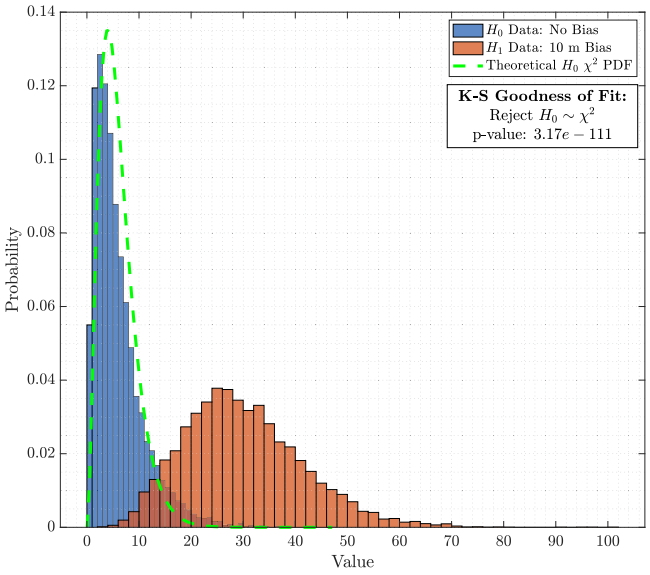

Fig. 9. Test statistic distributions: No-update sequence ($M = 3$, $\Delta t_M = 20$ s).

V. CONCLUSION

This article has proposed a novel method for real-time model validation for plug-and-play sensors, specifically aimed at all-source navigation systems. The proposed method, referred to as RVPS, enabled, through an innovative use of the Kalman–Schmidt partial update, the estimation of sensor-unique states without compromising the navigation solution, thereby protecting the integrity of the navigation solution during the validation period, all using a single existing filter. A series of Monte–Carlo simulations demonstrated the method's ability to not only detect invalid sensor models more reliably when compared to traditional methods such as horizontal position solution separation and residual sequence monitoring, but, additionally, prevent the detection process from corrupting the navigation solution. For brevity, the research presented in this article focused on detecting an unmodeled bias. Therefore, though not analytically conclusive across all fault types and use cases, this novel application of the partial update has been shown to outperform solution separation and conventional residual sequence monitoring in at least one example use case, where all the detectors were evaluated equally during the given scenario. This method complements previous developments in all-source APNT FDE such as [22], and directly enables self-correcting plug-and-play open architecture navigation systems such as [21]. Future work in this area will focus on experimental validation of the proposed method as well as the development of an online method of residual sequence spacing optimization.

REFERENCES

[1] D. T. Venable
Improving real world performance of vision aided navigation in a flight environment
Tech. Rep., Air Force Institute of Technology, WPAFB, 2016.

[2] J. Curro and J. Raquet
Navigation using VLF environmental features
In *Proc. IEEE Position Location Navigation Symp.*, 2016, pp. 373–379.

[3] A. Canciani and J. Raquet
Absolute positioning using the Earth's magnetic anomaly field
Navigation, vol. 63, no. 2, pp. 111–126, 2016.

[4] D. A. Grejner-Brzezinska, C. K. Toth, T. Moore, J. F. Raquet, M. M. Miller, and A. Kealy
Multisensor navigation systems: A remedy for GNSS vulnerabilities?
Proc. IEEE, vol. 104, no. 6, pp. 1339–1353, Jun. 2016.

[5] T. Kerr
Statistical analysis of a two-ellipsoid overlap test for real-time failure detection
IEEE Trans. Autom. Control, vol. 25, no. 4, pp. 762–773, Aug. 1980.

[6] B. Brumback and M. Srinath
A chi-square test for fault-detection in Kalman filters
IEEE Trans. Autom. Control, vol. 32, no. 6, pp. 552–554, Jun. 1987.

[7] M. A. Sturza
Navigation system integrity monitoring using redundant measurements
Navigation, vol. 35, no. 4, pp. 483–501, 1988.

[8] F. van Graas and J. L. Farrell
Baseline fault detection and exclusion algorithm
In *Proc. 49th Annu. Meeting Inst. Navigation*, 1993, pp. 413–420.

[9] B. W. Parkinson and P. Axelrad
Autonomous gps integrity monitoring using the pseudorange residual
Navigation, vol. 35, no. 2, pp. 255–274, 1988.

[10] Y. C. Lee
et al. Analysis of range and position comparison methods as a means to provide GPS integrity in the user receiver
In *Proc. 42nd Annu. Meeting Inst. Navigation*, 1986, pp. 1–4.

[11] R. S. Young and G. A. Mcgraw
Fault detection and exclusion using normalized solution separation and residual monitoring methods
Navigation, vol. 50, no. 3, pp. 151–169, 2003.

[12] M. Brenner
Integrated GPS/inertial fault detection availability
Navigation, vol. 43, no. 2, pp. 111–130, 1996.

[13] M. Brenner
Implementation of a RAIM monitor in a GPS receiver and an integrated GPS/IRS
In *Proc. ION GPS-90*, 1990, pp. 397–406.

[14] U. I. Bhatti, W. Y. Ochieng, and S. Feng
Integrity of an integrated GPS/INS system in the presence of slowly growing errors. Part I: A critical review
GPS Solutions, vol. 11, no. 3, pp. 173–181, 2007.

[15] U. I. Bhatti, W. Y. Ochieng, and S. Feng
Integrity of an integrated GPS/INS system in the presence of slowly growing errors. Part II: Analysis
GPS Solutions, vol. 11, no. 3, pp. 183–192, 2007.

[16] U. I. Bhatti
An improved sensor level integrity algorithm for GPS/INS integrated system
In *Proc. ION GNSS 19th Int. Tech. Meeting*, 2006, pp. 3012–3023.

[17] M. Joerger, F.-C. Chan, and B. Pervan
Solution separation versus residual-based RAIM
NAVIGATION: J. Inst. Navigation, vol. 61, no. 4, pp. 273–291, 2014.

[18] Ç. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan
An INS monitor to detect GNSS spoofers capable of tracking vehicle position
IEEE Trans. Aerosp. Electron. Syst., vol. 54, no. 1, pp. 131–143, Feb. 2018.

[19] C. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan
Sequential integrity monitoring for Kalman filter innovations-based detectors
In *Proc. 31st Int. Tech. Meeting Satell. Division Inst. Navigation, ION GNSS+ 2018*, 2018, pp. 2440–2455.

[20] C. Tanıl
et al. Optimal INS/GNSS coupling for autonomous car positioning integrity
In *Proc. 32nd Int. Tech. Meeting Satell. Division Inst. Navigation, ION GNSS+ 2019*, 2019, pp. 3123–3140.

[21] J. Jurado, J. Raquet, and C. M. Schubert Kabban
Autonomous and resilient management of all-source sensors for navigation
In *Proc. ION Pacific PNT Meeting*, Apr. 2019, pp. 142–159.

[22] J. D. Jurado, J. F. Raquet, C. M. S. Kabban, and J. Gipson
Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring
Navigation, 2020, doi: 10.1002/navi.384.

[23] J. D. Jurado and J. F. Raquet
Towards an online sensor model validation and estimation framework
In *Proc. IEEE/ION Position Location Navigation Symp.*, Apr. 2018, pp. 1319–1325.

[24] P. S. Maybeck
Stochastic Models, Estimation, and Control, vol. 2. Alexandria, VA, USA: Navtech, 1984.

[25] R. E. Kalman
A new approach to linear filtering and prediction problems
J. Basic Eng., vol. 82, no. 1, p. 35, 1960.

[26] P. S. Maybeck
Stochastic Models, Estimation, and Control, vol. 1. Alexandria, VA, USA: Navtech, 1982.

[27] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart
The Mahalanobis distance
Chemometrics Intell. Lab. Syst., vol. 50, no. 1, pp. 1–18, 2000.

[28] K. M. Brink
Partial-update Schmidt–Kalman filter
J. Guid. Control Dyn., vol. 40, no. 9, pp. 2214–2228, 2017.

[29] R. Y. Novoselov, S. M. Herman, S. M. Gadaleta, and A. B. Poore
Mitigating the effects of residual biases with Schmidt–Kalman filtering
In *Proc. 7th Int. Conf. Inf. Fusion*, vol. 1, p. 8, 2005.

[30] J. Dolloff
Kalman filter outputs for inclusion in video-stream metadata: Accounting for the temporal correlation of errors for optimal target extraction
In *Airborne Intell., Surveillance, Reconnaissance (ISR) Syst. and Appl. IX*, vol. 8360. International Society for Optics and Photonics, Baltimore, Maryland, United States, 2012, p. 83600I.

[31] J. Dolloff, B. Lofy, A. Sussman, and C. Taylor
Strictly positive definite correlation functions
In *Signal Process. Sensor Fusion, and Target Recognit. XV*, vol. 6235. International Society for Optics and Photonics, Orlando (Kissimmee), Florida, United States, 2006, p. 62351A.

[32] E. A. Wan and R. Van Der Merwe
The unscented Kalman filter for nonlinear estimation
In *Proc. IEEE Adaptive Syst. Signal Process., Commun. Control Symp.*, 2000, pp. 153–158.

[33] K. M. Brink
Unscented partial-update Schmidt–Kalman filter
J. Guid. Control Dyn., vol. 41, no. 4, pp. 929–935, 2017.

[34] S. M. Kay
Fundamentals of Statistical Signal Processing, Vol. II: Detection Theory. Prentice Hall, Upper Saddle River, New Jersey, United States, 1998.

[35] G. Casella and R. L. Berger
Statistical Inference, vol. 2. Pacific Grove, CA, USA: Duxbury, 2002.

[36] M. H. Kutner, C. Nachtsheim, and J. Neter
Appl. Linear Regression Models, vol. 4. McGraw-Hill/Irwin, Boston, Massachusetts, United States, 2004.

Juan Jurado received the B.S. degree in electrical engineering from Texas A&M University, College Station, TX, USA, the M.S. degree in electrical engineering from the Air Force Test Pilot School, Edwards, CA, USA, and M.S. in flight test engineering and Ph.D. degrees in electrical engineering from the Air Force Institute of Technology, Wright-Patterson AFB, OH, USA.

He is the Director of Education with the U.S. Air Force Test Pilot School. His research interests include aircraft performance modeling, online sensor calibration, image processing, visual-inertial navigation, and statistical sensor management for multi-sensor navigation problems.

John Raquet received the B.S. degree in astronautical engineering, M.S. degree in aero/astro engineering, Ph.D. in geomatics engineering. He is the Director of the Dayton Office of Integrated Solutions for Systems (IS4S), Beavercreek, OH, USA. He formerly was the Director of the Autonomy and Navigation Technology (ANT) Center, Air Force Institute of Technology, USA, where he was also a Professor of Electrical Engineering. He has been developing navigation system technology for more than 30 years.

Christine M. Schubert Kabban received the M.S. degree in applied statistics, Ph.D. degree in applied mathematics, MBA in finance. He is a Professor of Statistics with the Air Force Institute of Technology, Wright-Patterson AFB, OH, USA. She has been researching and practicing statistics for over 20 years in clinical, engineering, and statistical fields. Her research interests include applications to structural health monitoring and autonomous systems.